

**WHAT IS CLAIMED IS:**

- 1           1. A method comprising:  
2           adding a first plurality of data elements to a second plurality of data elements  
3                 generating a plurality of intermediate results;  
4           adding two of the plurality of intermediate results and repeating with different  
5                 combinations of the plurality of intermediate results generating a plurality  
6                 of sum results; and  
7           discarding the two least significant bits of each sum result of the plurality of sum  
8                 results.
- 9           2. The method as recited in Claim 1, further comprising:  
10           performing a carry in of a value of one when performing the adding the first  
11                 plurality of data elements to the second plurality of data elements.
- 12           3. The method as recited in Claim 1, further comprising:  
13           performing a carry in of a rounding term when performing the adding the two of  
14                 the plurality of intermediate results and the repeating.
- 15           4. The method as recited in Claim 3, wherein the rounding term is a variable  
16           capable of having a value of one and, at a different time, a value of zero.
- 17           5. The method as recited in Claim 1, further comprising:  
18           performing a carry in of a value of one when adding the first plurality of data  
19                 elements to the second plurality of data elements; and

20 performing a carry in of a rounding term when adding the two of the plurality of  
21 intermediate results and when repeating.

22 6. The method as recited in Claim 1, wherein the first plurality of data elements  
23 and the second plurality of data elements each comprise eight eight-bit unsigned data  
24 elements.

25 7. The method as recited in Claim 1, wherein the first plurality of data elements  
26 and the second plurality of data elements each comprise eight sixteen-bit data elements.

27 8. The method as recited in Claim 1, wherein the method comprises executing a  
28 Single-Instruction/Multiple-Data (SIMD) instruction.

29 9. The method as recited in Claim 1, wherein the method is performed utilizing  
30 Single-Instruction/Multiple-Data (SIMD) circuitry.

31 10. A method comprising:  
32 adding an  $i_{th}$  data element of a first source to an  $i_{th}$  data element of a second  
33 source creating an  $i_{th}$  intermediate result for  $i = 1$  to  $N$ , wherein  $N$  is an  
34 integer greater than 1;  
35 adding a  $j_{th}$  intermediate result to a  $(j+1)_{th}$  intermediate result creating a  $j_{th}$  sum  
36 result for  $j = 1$  to  $(N-1)$ ; and  
37 discarding two least significant bits of each  $j_{th}$  sum result.

38 11. The method as recited in Claim 10, further comprising:

performing a carry in of a value of one when adding the  $i_{th}$  data element of the first source to the  $i_{th}$  data element of the second source; and performing a carry in of a rounding term when adding the  $j_{th}$  intermediate result to the  $(j+1)_{th}$  intermediate result.

12. The method as recited in Claim 11, wherein the rounding term is selected from a group consisting of a value of one and a value of zero.

13. The method as recited in Claim 10, wherein  $N = 8$ .

14. The method as recited in Claim 10, wherein the method is performed during execution of a Single-Instruction/Multiple-Data (SIMD) instruction.

15. An apparatus comprising:

a plurality of first adders, each first adder of the plurality of first adders operative to add two operands of a plurality of operands into one of a plurality of intermediate results;

a plurality of second adders, each second adder of the plurality of second adders operative to add two intermediate results of the plurality of intermediate results into one of a plurality of sum results; and

discard circuitry operative to discard the two least significant bits of each sum result of the plurality of sum results.

57           16. The apparatus as recited in Claim 15, wherein the plurality of first adders  
58 comprises eight first adders and the plurality of second adders comprises seven second  
59 adders.

60           17. The apparatus as recited in Claim 15, wherein the discard circuitry comprises  
61 a plurality of shift registers.

62           18. The apparatus as recited in Claim 15, wherein each of the first adders are  
63 operative to add two eight-bit input operands producing a nine-bit intermediate operand  
64 and each of the second adders are operative to add two nine-bit intermediate operands  
65 producing a ten-bit output operand.

66           19. The apparatus as recited in Claim 15, wherein each of the first adders are  
67 operative to add two sixteen-bit input operands producing a seventeen-bit intermediate  
68 operand and each of the second adders are operative to add two seventeen-bit  
69 intermediate operands producing an eighteen-bit operand.

70           20. The apparatus as recited in Claim 15, wherein routing of the plurality of  
71 operands and the plurality of intermediate results to the plurality of first adders and the  
72 plurality of second adders is selected according to microcode identified by a Single-  
73 Instruction/Multiple-Data (SIMD) instruction.

74           21. The apparatus as recited in Claim 15, wherein routing of the plurality of  
75 operands and the plurality of intermediate results to the plurality of first adders and the

plurality of second adders is selected according to decode logic and a Single-Instruction/Multiple-Data (SIMD) instruction.

22. The apparatus as recited in Claim 15, wherein the plurality of first adders, the plurality of second adders, and the discard circuitry form a Single-Instruction/Multiple-Data (SIMD) instruction execution circuit.

23. An apparatus comprising:

a plurality of first adders operative to add an  $i_{th}$  data element of a first source to an  $i_{th}$  data element of a second source generating an  $i_{th}$  intermediate result for  $i = 1$  to  $N$ , wherein  $N$  is an integer greater than 1;  
a plurality of second adders operative to add a  $j_{th}$  intermediate result to a  $(j+1)_{th}$  intermediate result generating a  $j_{th}$  sum result for  $j = 1$  to  $(N-1)$ ; and  
circuitry operative to discard two least significant bits of each  $j_{th}$  sum result.

24. The apparatus as recited in Claim 23, wherein the circuitry comprises a plurality of shift registers.

25. The apparatus as recited in Claim 23, wherein routing of the plurality of operands and the plurality of intermediate results to the plurality of first adders and the plurality of second adders is selected according to microcode identified by a Single-Instruction/Multiple-Data (SIMD) instruction.

26. A method comprising:

decoding an instruction identifying an averaging operation;

executing the instruction on a first source and a second source, wherein the first source comprises a first plurality of data elements and the second source comprises a second plurality of data elements; and storing a result, wherein the result comprises a third plurality of data elements; wherein the executing the instruction comprises:

- adding successive ones of the first plurality of data elements to successive ones of the second plurality of data elements generating a plurality of intermediate results;
- adding two of the plurality of intermediate results and repeating with different combinations of the plurality of intermediate results generating a plurality of sum; and
- discarding the two least significant bits of each sum result of the plurality of sum results generating the result.

27. The method as recited in Claim 26, wherein the executing the instruction further comprises:

- performing a carry in of a value of one when adding the successive ones of the first plurality of data elements to the successive ones of the second plurality of data elements; and
- performing a carry in of a rounding term when adding the two of the plurality of intermediate results and when repeating.

28. The method as recited in Claim 27, wherein the rounding term is selected from a group consisting of a value of one and a value of zero.

118 29. An apparatus comprising:  
119 a coprocessor interface unit to identify an instruction for an averaging operation, a  
120 first source having a first plurality of data elements and a second source  
121 having a second plurality of data elements;  
122 an execution unit to perform the averaging operation on the first plurality of data  
123 elements and the second plurality of data elements; and  
124 a register to store a result having a third plurality of data elements;  
125 wherein the execution unit is operative to:  
126 add successive ones of the first plurality of data elements to successive  
127 ones of the second plurality of data elements generating a plurality  
128 of intermediate results;  
129 add two of the plurality of intermediate results and repeating with different  
130 combinations of the plurality of intermediate results generating a  
131 plurality of sum results; and  
132 discard the two least significant bits of each sum result of the plurality of  
133 sum results forming the result.

134 30. The apparatus as recited in Claim 29, wherein the execution unit is further  
135 operative to:  
136 perform a carry in of a value of one when adding the successive ones of the first  
137 plurality of data elements to the successive ones of the second plurality of  
138 data elements; and  
139 perform a carry in of a rounding term when adding the two of the plurality of  
140 intermediate results and when repeating.

141           31. The apparatus as recited in Claim 30, wherein the rounding term is selected  
142   from a group consisting of a value of one and a value of zero.

143           32. A data processing system comprising:  
144   an addressable memory to store an instruction for an averaging operation;  
145   a processing core coupled to the addressable memory, the processor core  
146           comprising:  
147           an execution core to access the instruction;  
148           a first source register to store a first plurality of data elements;  
149           a second source register to store a second plurality of data elements; and  
150           a destination register to store a plurality of results of the averaging  
151           operation;  
152   a wireless interface to receive a digital signal comprising a third plurality of data  
153           elements; and  
154   an I/O system to provide the first and second plurality of data elements to the first  
155           and second source registers from the third plurality of data elements;  
156   wherein the execution core is operative to:  
157           add successive ones of the first plurality of data elements to successive  
158           ones of the second plurality of data elements generating a plurality  
159           of intermediate results;  
160           add two of the plurality of intermediate results and repeating with different  
161           combinations of the plurality of intermediate results generating a  
162           plurality of sum results; and



163                   discard the two least significant bits of each sum result of the plurality of  
164                   sum results generating the plurality of results.

165           33. The data processing system as recited in Claim 32, wherein the execution core  
166 is further operative to:

167           perform a carry in of a value of one when adding the successive ones of the first  
168           plurality of data elements to the successive ones of the second plurality of  
169           data elements; and  
170           perform a carry in of a rounding term when adding the two of the plurality of  
171           intermediate results and when repeating.

172           34. The data processing system as recited in Claim 33, wherein the rounding term  
173 is a variable capable of having a value of one and, at a different time, a value of zero.

174           35. An article comprising a machine-readable medium that includes machine  
175 readable instructions, the instructions operative to cause a machine to:  
176           add a first plurality of data elements to a second plurality of data elements  
177           generating a plurality of intermediate results;  
178           add two of the plurality of intermediate results and repeating with different  
179           combinations of the plurality of intermediate results generating a plurality  
180           of sum results; and  
181           discard the two least significant bits of each sum result of the plurality of sum  
182           results generating a result.

183           36. The article as recited in Claim 35, the instructions further operative to:

184 perform a carry in of a value of one when adding the first plurality of data  
185 elements to the second plurality of data elements; and  
186 perform a carry in of a rounding term when adding the two of the plurality of  
187 intermediate results and when repeating.